The effect of the Distance in Pedestrian Detection

David Vázquez Bermúdez david.vazquez@cvc.uab.es
Computer Vision Center
Edifici O, Universitat Autònoma de Barcelona
08193, Bellaterra (Spain)

Advisors: Dr. Antonio M. López and David Gerónimo

Abstract

Pedestrian accidents are one of the leading preventable causes of death. In order to reduce the number of accidents, in the last decade the pedestrian protection systems have been introduced, a special type of advanced driver assistance systems, in witch an on-board camera explores the road ahead for possible collisions with pedestrians in order to warn the driver or perform braking actions. As a result of the variability of the appearance, pose and size, pedestrian detection is a very challenging task. So many techniques, models and features have been proposed to solve the problem. As the appearance of pedestrians varies significantly as a function of distance, a system based on multiple classifiers specialized on different depths is likely to improve the overall performance with respect to a typical system based on a general detector. Accordingly, the main aim of this work is to explore the effect of the distance in pedestrian detection. We have evaluated three pedestrian detectors (HOG, HAAR and EOH) in two different databases (INRIA and Daimler09) for two different sizes (small and big). By a extensive set of experiments we answer to questions like which datasets and evaluation methods are the most adequate, which is the best method for each size of the pedestrians and why or how do the method optimum parameters vary with respect to the distance.

Keywords: ADAS, Pedestrian detection, HAAR features, EOH and HOG descriptors.

Contents

1. Introduction

Motor vehicle collisions are one of the leading preventable causes of death. The total worldwide historical number of car accident fatalities is difficult to estimate, but, about ten million people become traffic causalities each year and two or three of them are seriously injured. For instance, in 2003 the European Union reported 150000 injured and 7000 deaths in road accidents in which cars collided with pedestrians and cyclists. Pedestrian run overs represent the second largest source of traffic-related injuries [1].

In order to reduce these road accidents, different types of protection systems such as seat belts, airbags or ABS appear. Recently new lines of research tend to elaborate more intelligent systems which are known as Advanced Driver Assistance Systems (ADASs). In this work we focus on a special type of ADAS, the Pedestrian Protection Systems (PPSs) where an on-board camera explores the road ahead for possible collisions with pedestrians in order to warn the driver or performing braking actions [2, 3].

From a computer vision point of view pedestrian detection is a challenging task. The main challenges rely on the pedestrian appearance variability due to clothes, poses or sizes and the context where they can be found, like different scenarios with cluttered background, under uncontrolled illuminations, with shadows, occlusions, etc. Also, PPSs work in dynamic scenes where pedestrians and vehicle are in motion, so, a high performance on time and robustness is required.

Pedestrian detection produced a vast interest over the last years in the computer vision community. Thus, many techniques, models, features and general architectures have been proposed. As these proposals use different architectures, databases and evaluation criteria it is difficult to compare and study them. Recently, a pedestrian detection survey has been presented [2] which proposes a general module-based architecture (Fig. 1) that simplifies the comparison between specific detection tasks. It also reviews different approaches with respect to the tasks defined in the proposed architecture.

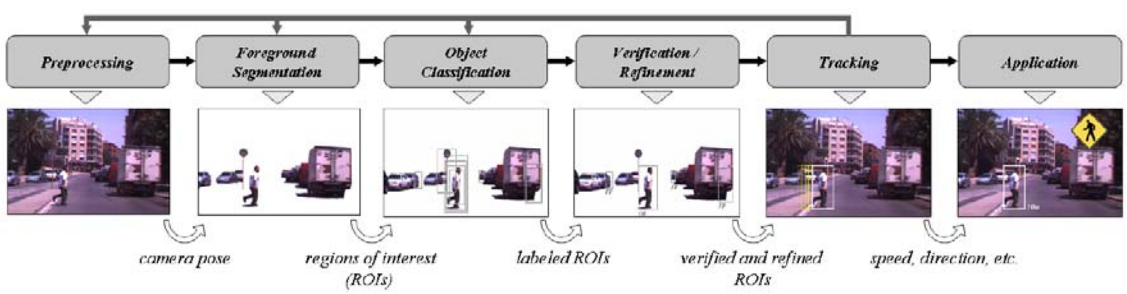


Figure 1: The general module-based architecture in [2] covers the structure of most of the systems. It is composed by six modules: preprocessing, foreground segmentation, object classification, verification, tracking and application.

Performing an extensive review of the related literature is not the aim of this work because of the lack of space and because there exist two recent good surveys in the literature [2, 3]. On the contrary, we focus on the object classification stage of the aforementioned architecture. This module receives a set of Regions Of Interest (ROIs) to be classified as pedestrians or non-pedestrians. Among the different methods proposed in this stage we focus on the appearance based ones, which define a space of image features and then train a learning machine on examples to obtain a classifier that is able to classify new samples.

Several learning machines have been used in the literature, but we can condensate the most important ones into three groups. Neural Networks (NN) [4] is a bio-inspired architecture based on layers of neurons that leads to a non-linear classifier. Support Vector Machines (SVM) [5] is a statistical method that finds a decision boundary by maximizing the margin between the different classes. Adaptive Boosting (AdaBoost) [6] builds a strong classifier by a combination of weak classifiers.

Several feature spaces or descriptors have been proposed in the literature as well. The simplest features were proposed by Gavrila et al.[7] which used grey scale image pixels with a NN-LRF as a learning machine, then Zao et al. [8] used image gradient magnitudes combined with a NN. Papageorgiou et al. [9] introduced the Haar wavelets features that compute the pixel difference between two rectangular areas in different configurations and can be seen as large scale derivatives; they used a SVM as for the classification. Viola and Jones [10] proposed an extended set of Haar wavelets features combined with an AdaBoost cascade. Gerónimo et al. [11] combined the Edge Orientation Histograms (EOH) with Haar wavelets in an AdaBoost cascade, resulting a robust and fast pedestrian detector. Dalal et al. [12] presented the Histogram of Oriented Gradients (HOG), a Shift [13] inspired feature that combined with a SVM is the reference on the state-of-the-art of pedestrian detection. Recently, new approaches overcoming the state-of-the-art appeared in the literature. For instance Tuzel et al. [14] propose a novel algorithm based on the covariance of several measures as features and LogitBoost and Riemannian manifolds to classify them. Felzenszwalb et al. [15] present an approach based on Dalal's HOG detector that consists of a representation of the whole pedestrian and several representations of pedestrians parts. The classification is done using latent SVM. It's currently one of the best methods for object detection.

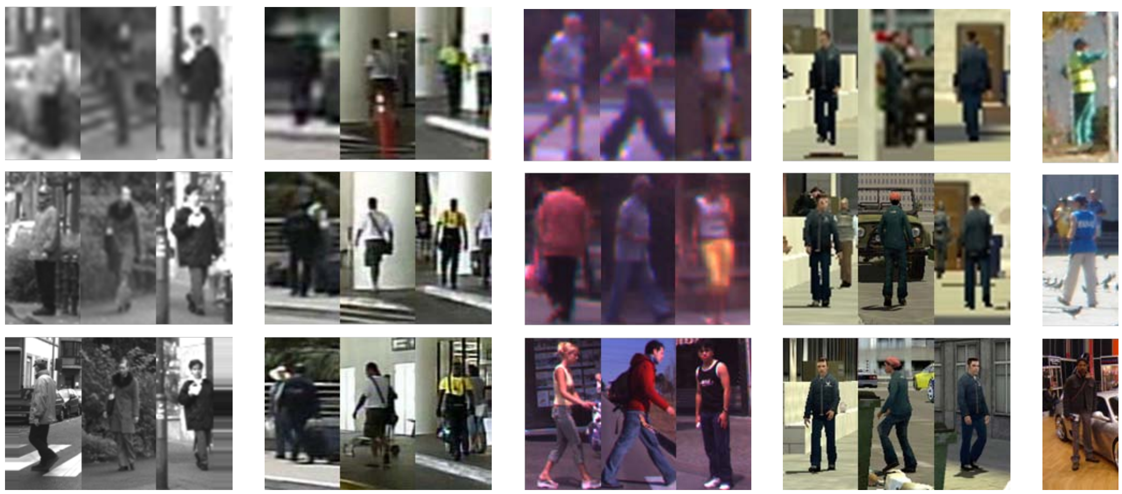
Geronimo et al. [2] conclude their survey by explaining the needs in PPS. The most important one seems to be the lack of good databases and benchmarking protocols. The authors also suggest that the effect of the distance and the detection of partially occluded pedestrians are important areas of research. The aim of this Master Thesis is to explore these two PPSs needs. During the course of this project, the idea of exploring the effect of the distance has been reinforced by Enzweiler and Gavrila's work [3].

More specifically, by this work we want to answer the following questions: (1) which are the most adequate datasets from the latest ones? (2) for detecting far away people, which is the difference between training a system with actual small pedestrians and training it with scaled pedestrians? (3) which is the best detection method for each size of the pedestrians and why? (4) how do the optimum parameters of each method vary with respect to the distance?

The remainder of this paper is organized as follows. In Sect. 2 we explain why the effect of the distance in pedestrian detection is of interest, and we overview the benchmark datasets and detection approaches needed to study such effect. The experiments, the evaluation criteria and the results are explained in the Sect 3 which finalizes with a discussion of the results. Finally, in Sect 4 we summarize the conclusions of this work, and draw some future work.

2. The effect of the distance in pedestrian detection

Given that pedestrians closer to the camera are seen with more detail than the ones which are further away, a study on the benefices of training different classifier models depending on the target distance is of key interest. In Figure 2 we can observe how the appearance of pedestrians varies significantly as a function of distance. This leads us to think that a system based on multiple classifiers, each specialized on different depths is likely to improve the overall performance with respect to a typical system based on a single general detector.



(a) Daimler09 (b) CALTECH (c) CVC (d) Virtual (e) INRIA

Figure 2: As can be seen from different datasets the appearance of pedestrians in these cases change dramatically with the distance. Up: Far; middle: medium distance; bottom: near.

Since distant pedestrians are smaller and have less details, they tend to be more difficult to classify. On the one hand, closer targets present more details and their classification is easier but, on the other hand, the latency of the system for detecting them must be lower than for far away ones. Then, we could build a system with a single classifier specialized on far pedestrians that provides interesting targets to be tracked[1] and a robust classifier that only with few frames could detected close pedestrians. In a second step, the outputs of all classifiers could be merged to make a decision based on a distance criterion, i.e. to yield the final classification result. However, integrating the trained classifiers in a real system with tracking it is out of the scope of this Master Thesis.

As far as we are concerned, the only existing references on this effect are the suggestion of Gerónomio et al. [2] of studying it and the experiments that have been published during the progress of this Master Thesis in the work of Enzweiler and Gavrila [3]. In this work, the authors evaluate the performance of three different detectors trained for far and near pedestrians to asses which method is the best for each distance but without looking for the best set of parameters.

1. With this tracking information we would have 2D image features over time that could be useful for temporal coherence analysis. Also it can be used to compute movement features.

In this work we train two pedestrian detectors (HOG and HAAR+EOH) in two dataset (INRIA and Daimler09) and for two different sizes (small and big) with the parameters proposed by the authors. Then, we evaluate each trained detector with two different techniques (per-window and per-image) for a more precise evaluation. Also we tune the parameters of the detectors for each particular case. In the next subsections we describe the different datasets and explain the pedestrian detection methods used.

2.1 Selected pedestrian benchmark datasets

Existing datasets can be grouped in three types: (1) *unconstrained person* datasets that contain people in a wide range of poses and occlusions, (2) *person* datasets that contain non-occluded people in different poses and backgrounds but with a restricted point of view and, since two months ago, (3) *pedestrian* datasets that contain upright or partially occluded pedestrians in an urban environment and usually with motion information and more complete labellings.

(a) INRIA (b) Daimler09 (c) CALTECH

Figure 3: Images from the selected datasets. The first row shows some images with pedestrians. The second row shows some cropped pedestrians sorted by their sizes.

The most important *unconstrained person* dataset is the PASCAL VOC [16], but this dataset is not useful for pedestrian detection because it contain persons in very different poses as lay over the floor or sit at a desk and also faces or other body parts that are not useful for pedestrian detection. Among the *person* datasets it is important to cite the MIT [9] and USC [17, 18] ones which nowadays are perfectly classified by the state-of-the-art, and the INRIA dataset [12] that remains the most widely used. Recently, some *pedestrian* datasets appeared. Among these databases it is important to cite the CVC [11] and the Caltech databases [19] and, finally, the Daimler06 [20] and the Daimler09 [3] ones.

Table 1, adapted from [19], provides a detailed overview of the existing datasets. Some datasets have thousands of images from only a few different tracked pedestrians so, in the table the number of different pedestrians have been also included by us after a close look to the datasets. Also, the Daimler09 dataset have been included. We can observe that the new *pedestrian* datasets (Daimler09 and Caltech) have significantly more examples of different sizes than the others. However, in Caltech only a few of these examples are useful for training, while the testing set is not publicly available. From Figure 3 we can see that the Caltech's pedestrians are very small and poorly labeled, which makes its use difficult for training. Accordingly, we discard the Caltech dataset.

	Training			Testing			Height			Properties					
	# pedestrians	# neg. images	# pos. images	# pedestrians	# neg. images	# pos. images	10% quantile	median	90% quantile	color images	per-image ev.	no selec. bias	video seqs.	temporal corr.	occ. labels
MIT [9]	924	-	-	-	-	-	128	128	128	✔					
USC [17, 18]	-	-	-	816	-	359	69	99	135		✔				
INRIA [12]	1208	1218	614	566	453	288	139	279	456	✔					
CVC [11]	700	6.1k	-	300	-	-	46	83	164	✔		✔			
Caltech [19]	192k	61k	67k	155k	56k	65k	27	48	97	✔	✔	✔	✔	✔	✔
	(845)		(?)	(?)	(?)	(?)	27	48	97						
Daimler09 [3]	3915	6.7k	-	56k	7.5k	14k		72			✔	✔	✔	✔	✔
				(259)		(1.9k)		72							

Table 1: Datasets comparison. The first six columns are the amount of training/testing data ($1k = 10^3$). The columns are: Number of unique Bounding Boxes (BB) labeled, number of images not containing any pedestrian and number of images containing at least one pedestrian. For the video sequenced based databases it is indicated in brackets the number of different and really useful pedestrians. The next three columns show the range in the scales of the images. The final columns summarize additional properties. It is important to notice that in the Caltech database the test is not publicly available.

In this work we are going to use the INRIA dataset because it is a spread reference in pedestrian detection and we want to compare it with other databases. From the *pedestrian* datasets, we discarded the CVC and Daimler06 ones because we are already familiar with them and we prefer to explore the new datasets of Daimler09. Daimler09 is the appropriate dataset for studying the effect of the distance given that it has a lot of examples from different sizes to train and also provides a video sequence fully annotated that allows us to evaluate the experiments.

2.2 Selected pedestrian detection approaches

We have selected two pedestrian detectors to evaluate the effect of the distance: Haar wavelet's (HW) and Edge Orientation Gradients (EOH) combined with AdaBoost, and Histogram of Oriented Gradients (HOG) combined with a linear SVM. These approaches are used in a sliding windows fashion and not integrated any 3D or tracking information. Besides the selected methods, there exist many other interesting approaches that could be applied [2] but the ones that we have chosen are the most representatives.

Our experimental setups are of two kinds: (1) Assign the detector parameters (i.e. sample resolution, feature size, etc) to the values reported by the original publications and (2) optimize these parameters in each dataset and distance. Let us explain the selected detectors and their underlying parameters.

2.2.1 Histogram of oriented gradients

Dalal et al. [12] proposed a pedestrian detector based on Histogram of Oriented Gradients (HOG) features inspired on SIFT and a SVM learning machine. It is the reference in the state-of-the-art of pedestrian detection. These features model the shape and appearance using normalized histograms of the image gradient orientation. The idea is to divide the image with a dense spatial grid in small regions called *cells*. A cell is represented as a histogram of its local gradients binned according to their orientation and weighted by their magnitude. These cells are grouped in larger regions called *blocks*. A block is represented as a feature vector formed by concatenated and normalized histograms of its cells. The final descriptor is a feature vector formed by all the blocks attached and it is classified using a linear SVM.

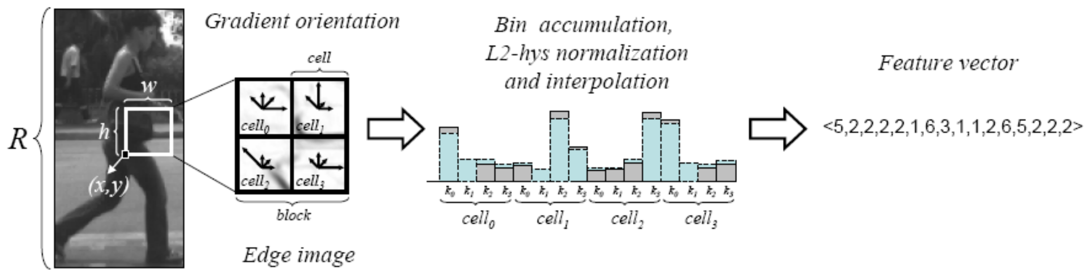


Figure 4: Histogram of Oriented Gradients.

The training process consists in computing the features from the training examples and train a SVM to obtain a first classifier. Then the training images are scanned with this trained classifier in a sliding window fashion to obtain false positives that are added to the initial training set. This new dataset use to be formed with more complicated samples and a new classifier is trained with it. This new training step is what we call bootstrapping and it can be done several times until the performance of the classifier does not improve anymore.

To compute the features we use the parameters suggested by the authors: a canonical window size of 64×128 pixels, a simple 1D mask without any smoothing is used to compute the gradient, each cell is a region of 8×8 pixels represented by a histogram of 9 orientation bins in the range $[0, 180]$ degrees, blocks of 2×2 cells that have an overlapping of 50% and normalized using L2-Hys. Finally, it is classified with a linear SVM with cost $C = 0.01$. Then for each dataset we try to optimize these parameters.

2.2.2 Haar wavelets and edge orientation gradients

This classifier, which consists of Haar wavelets and Edge Orientation Histograms (HaarrEOH) as features and Real AdaBoost as learning machine, have been originally proposed by Levi and Weiss to perform face detection in [21]. Then in [11] Gerónimo et al. add some slight modifications and use it to classify pedestrian samples.

Haar wavelets, introduced by Papageorgiou et al. [9], are a set of filters that compute the pixel difference between rectangular areas in different configurations and they can be understood as region derivatives. These features can be computed efficiently by means of the Integral Image representation (II), and computed by four II accesses. The original set of features is formed by three filters (horizontal, vertical and diagonal) but we use the extended set proposed by Viola and Jones [10] showed in Figure 5.

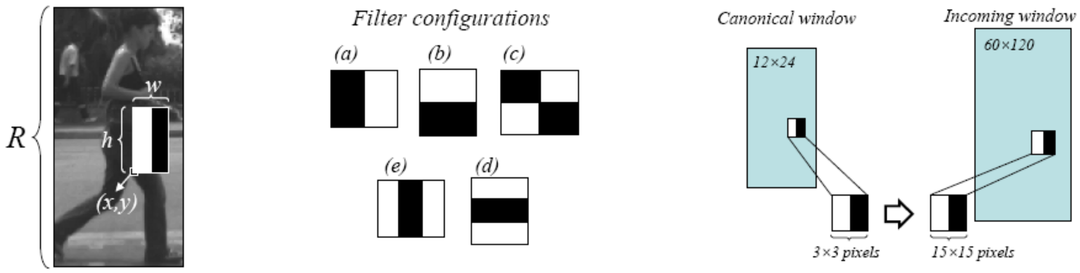


Figure 5: Haar wavelets features.

Edge Orientation Gradients (EOH) features, introduced by Gerónimo et al. [11] for pedestrian detection, are based on the strong edge information of the image and they are invariant to global illumination changes. These features compare the ratio between two different orientations in a region of the image and it is also possible to compute them in a efficient way by using the integral image.

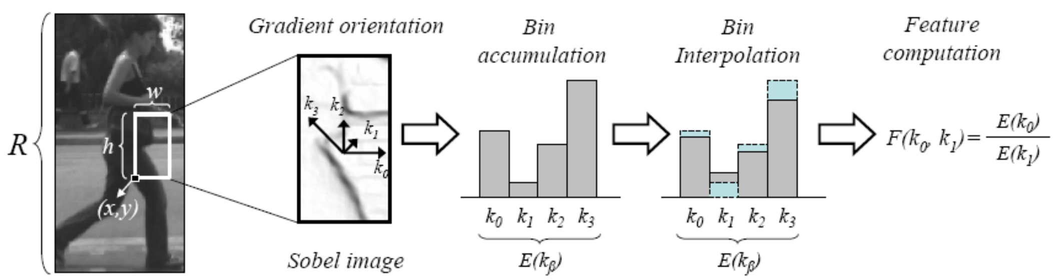


Figure 6: Edge Orientation Gradients.

The parameters used as default are four bins for the EOH and some constrains in the number of possible Haar and EOH filters used, so a feasible training is possible. We compute all the possible filters in a canonical window of 12x24 pixels with sizes from 2x2 pixels to 12x24 pixels with a step of 2 pixels of size between one filter and the next and without any restriction in the position. With these restrictions we obtain about 80 000 different features for each window.

The learning machine used is the RealAdaBoost because of the huge number of features. AdaBoost can learn which among all these features are the most discriminative and train a classifier only with a subset of these features. This training requires a large computational cost and memory but the trained classifier is very fast because only some of a subset of the filters have to be computed. Also, it can be speed up with the use of the cascades where the easy windows can be discarded in early stages using a very small number of features. Instead of using the cascade, we use a single AdaBoost trained on the bootstrapping fashion, as it have been done with the SVM in Dalal's HOG, because this training is faster than training several cascades.

In order to analyze windows of different sizes, a spatial normalization is required to establish an equivalence between the features computed in each window. To achieve that, this method does not explicitly resize the windows given that features can be computed in a way that it is equivalent to resizing them but more efficiently. We introduce a modification in the method: the image pyramid representation, that explicitly resize the windows. As it will be seen in the experiments, although this modification increases the computational cost, it also improves the classifier performance.

3. Experiments

Our experiments consist in evaluating the mentioned classifiers (HOG and HaarEOH based) in the selected databases (INRIA and Daimler09) and for two different sizes (small and big) with the authors' proposed parameters. After, we will also tune the parameters of the detectors for each concrete case.

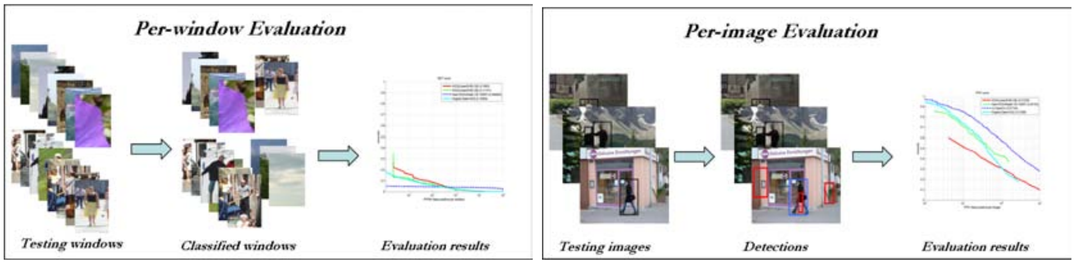
The total processing time needed to train, test, and evaluate these experiments is about one month of CPU time on a 2.83 GHz Intel processor and one week of a Xeon server, using our own optimized implementations in C++. For instance, to train and evaluate a HOG there is need about one day of CPU for the INRIA dataset and two days for the Daimler09. And, more complicated is to train a HaarEOH that needs one day of a Xeon Quad core server in order to learn about 500 features. Looking at the testing time, for instance to detect the big pedestrians in the Daimler09 dataset with the HOG method it spends about 30s/frame while HaarEOH spends 3s/frame which implies an speed up of ten times.

Because of the huge number of features used in the HaarEOH (80 000) and the great number of examples (15 000 positive samples) of the Daimler09 dataset, it have not been feasible to train with all the data, so this dataset have been reduced for this method to 2 400 positive samples. Also, to speed up the training process, the number of features used for this method is very low (100 features in Daimler and 500 on INRIA datasets while HOG uses about 3 000 features) and some experiments have been trained without the bootstrapping stage. Then, for this dataset the comparison of HOG vs HaarEOH is not totally fair from the very begging. How ever our main interest now is the effect of the distance, not comparing the HOG vs HaarEOH.

3.1 Evaluation methodology

Before going into the details of the experiments let us explain the evaluation methodology. The selected pedestrian detectors are sliding window based. These detectors are densely scanned at several scales across the image and finally the detections are combined using a Non-Maximum Suppression (NMS) procedure. To evaluate them, there exist two established methodologies. The most widely used is the *per-window* performance introduced by Dalal et al. [12] and the other one is the *per-image* evaluation as used for example in Pascal dataset [16].

In the per-window approach (Fig. 7a), the detector is evaluated by classifying cropped pedestrians versus random crops from negative images avoiding to densely scan the image and the NMS. The performance is mainly plotted in two different curves: The Receiver Operating Characteristic (ROC) curve plots the recall versus the false positive rate (or fall-out); and the Detection Error Trade off (DET) curve plots the miss-rate versus the False



(a) Per-Window (b) Per-Image

Figure 7: Evaluation methodologies.

Positive Per analyzed Window (FPPW). As the DET curve is more clear for the comparison of different detectors and it is the most widely used in the references, it is the one we use for the per-window evaluation.

The problem of the per-window evaluation is that it avoids the errors caused by the sliding window step, like detections at imprecise scales and positions or false positives over body parts. Also the NMS and its interactions with the position and scale of the detected windows are not taken into account.

In the per-image approach (Fig. 7b), an image is given to the detector and it returns a list of Bounding Boxes (BB) with a given confidence. In this case, the detector has to performing a sliding window and a NMS. The evaluation consists in performing a correspondence between the BB detections BB_{dt} and the BB groundtruth BB_{gt}. Two BBs form a potential matching if they overlap sufficiently so the standard measure is the overlapping coefficient (Eq. 1) used in the PASCAL Challenge [16]. Also, this correspondence can be performed one-to-one which is the most commonly used [12] or many-to-many, more focused on real applications [3]. To compare methods there are also two different curves: The Precision-Recall (PR) and the False Positives Per Image (FPPI).

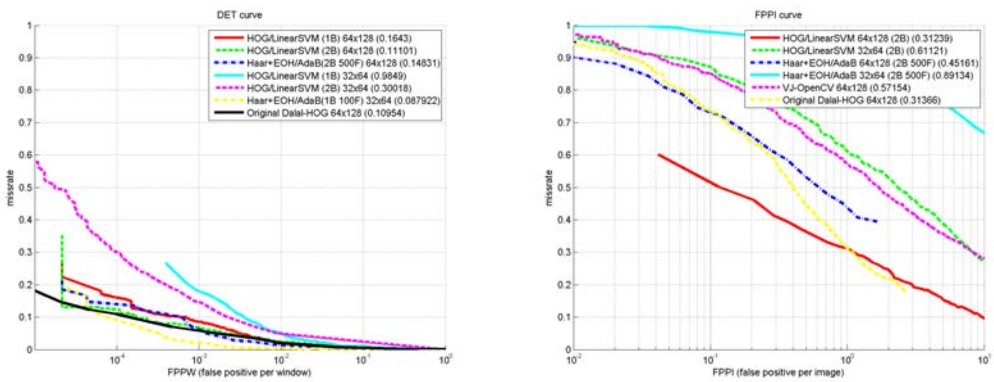
$$\Gamma(a_i, a_j) = \frac{A(a_i \cap a_j)}{A(a_i \cup a_j)} \tag{1}$$

In the literature the per-window evaluation is performed by most of the authors in the same way but it is not clear how the random crops from the negative images are extracted. However, in the per-image evaluation we can find differences between the authors like the overlapping coefficient (0.5 for INRIA and 0.25 for Daimler09) and the matching correspondence (one-to-one in INRIA dataset and many-to-many in Daimler09). In the one-to-one case, each BB_{dt} and BB_{gt} may be matched at most once and it counts as a detection while unmatched BB_{dt} or BB_{gt} count as false negatives. In the many-to-many case, each BB_{gt} can be matched with several BB_{dt} and it counts as only one detection, and the unmatched BB_{dt} or BB_{gt} count as false negative. An important detail is the notion of optional groundtruth BB_{opt} (i.e. occluded, very small pedestrians, persons on bikes, etc), the BB_{dt} matched with the BB_{opt} do not count as detection and the unmatched BB_{opt} do not count as false negative.

The typical assumption is that better *per-window* performances will lead to better scores on per-image evaluation; however, in practice it can fail because per-window neither takes into account the localization or scaling of the BBs nor the further steps like NMS. However, the per-window evaluation is faster to compute and gives a first idea of the overall performance of the system.

3.2 Results

The goal of the first set of experiments is to validate that the implemented algorithms performs correctly. For this propose we use the INRIA dataset that allow us to compare our results with the obtained by other authors. Usually in the per-window evaluation to compare the results of two methods we look the missrate value at FPPW of 10^{-4} over the DET curves and in the per-image evaluation we compare the missrate values at 1 FPPI.



(a) Per-window evaluation: DET

(b) Per-image evaluation: FPPI

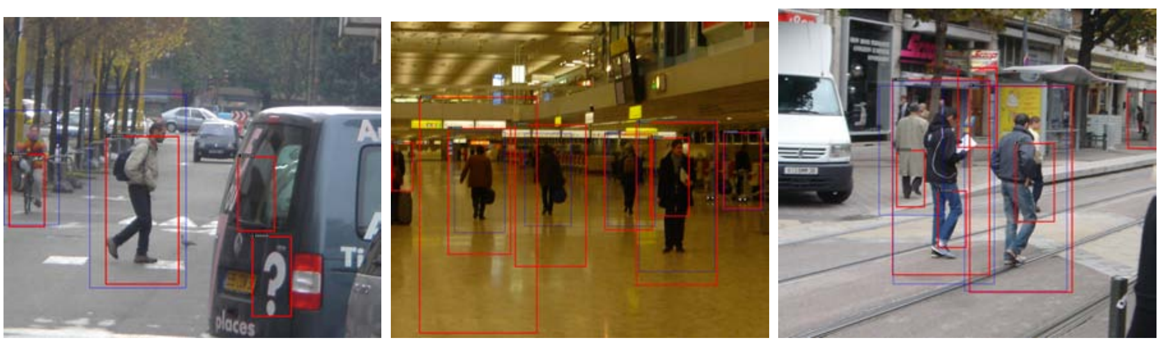
Figure 8: Performance evaluation of the HOG and Haar+EOH methods in the INRIA dataset. In the plot legend is indicated the number of bootstrapping iterations (1B or 2B), the number of features used in the AdaBoost (100F or 500F), the size of the training samples (32x64 or 64x128) and the real number inside the parenthesis is the missrate value at 10^{-4} FPPW in the per-window evaluation and at 10^{0} FPPI in the case of the per-image evaluation.

In these experiments the methods have been trained with 2400 samples of pedestrians and 12000 cropped windows extracted from 1200 negative images as proposed by Dalal [12]. Then, a further step of bootstrapping have been done without any restriction in the number of false positives to include in the train. For the per-window test, 1200 samples of pedestrians and 10^5 negative windows have been used. For the per-image evaluation 288 images have been used following the one-to-one matching criteria with an overlapping coefficient of 0.5 as proposed by Dalal.

Figure 8 shows the obtained performance of the proposed methods on INRIA dataset and it can be compared with the original results obtained by their authors. In the DET curve we can see that our implementation of HOG gives nearly the same results than the original HOG of Dalal and our implementation of HaarEOH gives similar results. In the FPPI curve we can see that in 1 FPPI the original HOG gives the same results than our implementation HOG but the HaarEOH is now worse. But, if we compare the missrate for more strict FPPI our implementations perform better. In addition, we can compare the

Haar method with HaarEOH and we can see that combination of Haar with EOH improve the results considerably. From here we can conclude that for this dataset the best algorithm is the HOG and that the methods are well implemented.

(a) HOG



(b) HAAR+EOH

Figure 9: Some detections of the HOG and Haar+EOH methods in the INRIA dataset. The blue BBs are the groundtruth and the red ones are the detections.

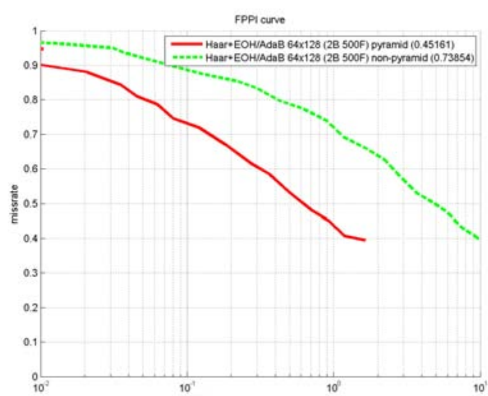


Figure 10: Difference between the use of the image pyramid representation versus scaling the features.

In the original HaarEOH implementation the windows to analyze are not resized. Instead, the features are computed in a way that it is equivalent to resizing the window but more efficient. We introduced a modification in the method, the image pyramid representation, that explicitly resizes the windows. In order to see this effect we show in Figure 10 how the pyramid improves the classifier performance.

In Figure 9 we can see some images of detections that ilustrate the differences in the detections and in Figure 11 we can see the obtained models for each detector. In the HOG's model we can see the weights learned by the SVM to discriminate the pedestrians from the background. In the HaarEOH model we can see the first features that the AdaBoost learned as the most discriminative ones.

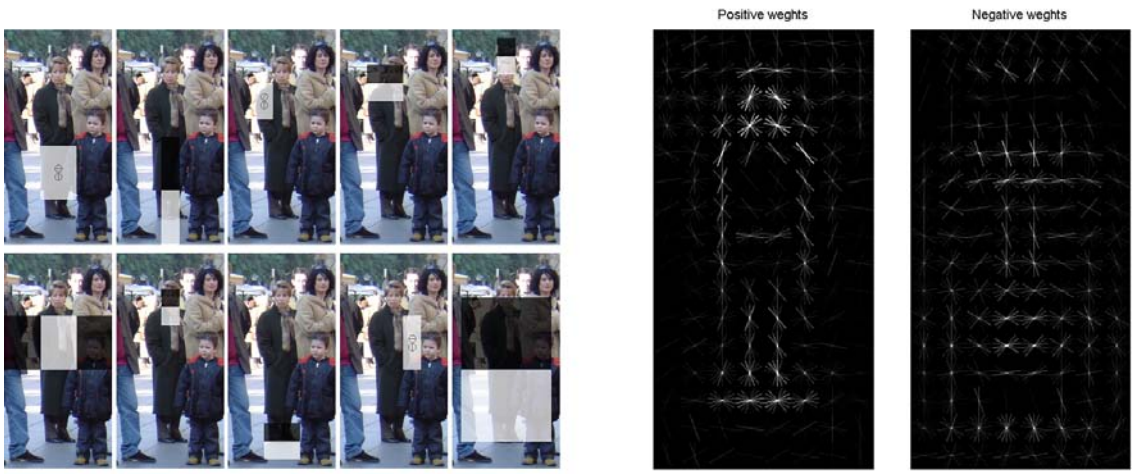
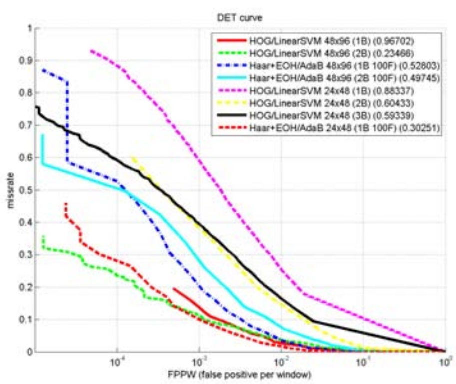


Figure 11: Models learned by the HaarEOH and HOG methods in the INRIA dataset.

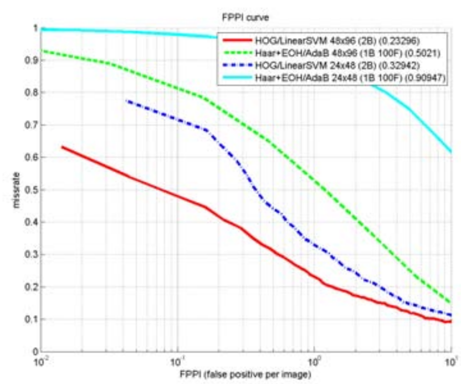
Now that we know that our algorithm implementations performs as expected over the INRIA datasets, we evaluate them over the Daimler09 one. For this purpose we train the methods on the big images (48x96 pixels) of training and we test them as it was proposed in [3] with pedestrians of a height over 72 pixels. In [3] three methods (Haar, HOG and NN-LRF) have been employed and the authors study the effect of three parameters. These parameters are external to the methods and common for all of them, these are: (1) The spatial stride and scale steps of the sliding window, (2) the number of bootstrapping or retraining iterations needed to obtain the final trained models and (3) resolution of the training images. As shown in [3], increasing the number of bootstrapping iterations and decreasing the scale step and spatial stride, improves the performance results. However, it also increases the computational cost and they expend several months of CPU time to obtain the results. Here, we want to evaluate the behavior of the methods while changing their internal parameters. For this purpose we fix an small number of bootstrapping iterations (1 iteration) and a big spatial grid (strideX=0.16, StrideY=0.08 and scaleStep=1.20) to make the computation feasible in time.

In these experiments the HOG method have been trained with 15 000 samples of pedestrians and 12 000 cropped windows extracted from 6 000 negative images as proposed by Enzweiler and Gavrila [3]. Then, a further step of bootstrapping have been done with the restriction of including only 2 false positives from each frame. For the per-window test, we have extracted 1 500 samples of pedestrians and 10^5 negative windows from the test video of the dataset. For the per-image evaluation we do not use the complete video of testing because it is too long (22 000 frames) and in many of the frames there are not pedestrians. So, we have extracted 1 000 frames where there is at least on pedestrian. The matching criteria is many-to-many with an overlapping coefficient of 0.25 and the optional pedestrians (occluded, small, etc) are not taken into account either as false positives either false negatives as proposed by Enzweiler and Gavrila. The HaarEOH method uses the same configuration but with 2 400 positive samples.

In Figure 12 it can be seen that again the HOG method performs slightly better than the other. In Figure 13 we can see some images of detections where we can see the differences in the detections. The results of HOG over Daimler dataset in the per-image evaluation are comparable to the obtained results in the original paper but we cannot show them because we do not have the original data to plot together as in the INRIA case. Again we can see that although the HaarEOH performs similar than HOG in the per-window evaluation, it performs worse in the case of per-image.



(a) Per-window evaluation: DET



(b) Per-image evaluation: FPPI

Figure 12: Performance evaluation of the HOG and Haar+EOH methods in the Daimler09 dataset.

Next we evaluate the effect of the distance in pedestrian detection. To evaluate this we scale the images of the two datasets to a smaller size: 32x64 for INRIA and 24x48 for Daimler09. Then we train and test in the per-window evaluation over the scaled images. And in the per-image evaluation we use the classifiers trained with the small pedestrians to detect the big ones in order to compare the classifiers over the same set of images. Now, we can compare the performance of the classifier trained with different sample sizes. Analyzing Figures 8 and 12 it can be appreciated that for the HOG detector the bigger the image is the better the detector performs. In the case of HaarEOH, surprisingly, the detector performs better with the smaller images. However, again the HaarEOH gives bad results for the per-image evaluation. Figure 14 shows the per-image evaluation over the small pedestrians from 24x48 to 48x96 pixels of size next experiments we will test.

In the previous experiment the small pedestrians used for training have been scaled from big pedestrians. Now, we want to evaluate if there is any difference between training

(a) HOG

(b) HAAR+EOH

Figure 13: Some detections of the HOG and Haar+EOH methods in the Daimler09 dataset.

with these scaled pedestrians and training with actual small pedestrians. The only small pedestrians that we have are the test images from the Daimler09 dataset. These are 270 different pedestrians tracked along time that suppose about 4500 images. We split this data in two sets: one of 2400 images for training and another of 2100 for testing. It is important to see that as these images are extracted from only 270 pedestrian, there are many images that are very similar. We train one HOG classifier with this images and another with the first 2400 scaled images of the training set. Then we evaluate both algorithms with the new test set. Looking at Figure 14 we can see that the obtained results with the actual small images is slightly better. We expect that if we would have actual small pedestrians extracted from more different persons this difference in the results could be much bigger.

Once we have seen how the algorithms perform for each dataset with the original parameters, we can optimize these parameters in order to try to adapt the methods for the dataset. In the HaarEOH the only internal parameters are the number of bins of the EOH and the number of different locations and sizes of the filters. The number of filters for all the cases is never greater than 80 000 because otherwise it would not be feasible for training, so, we do not change this parameter. In the case of the HOG we expect that for small images a minor number of cells by block or a minor number of bins in the histograms could improve the results.

The parameters tuning of the HOG has been done in the same way as Dalal et al. suggest in [12]. There several parameters are optimized: the gamma normalization, the gradient computation, the spatial blocks and sizes, the binning orientations and the normalization schemes. We optimize only the size of the cells and blocks and the orientation bins. In figures 15a and 15c we can see the missrate for a FPPW of 10^{-3} for different configuration

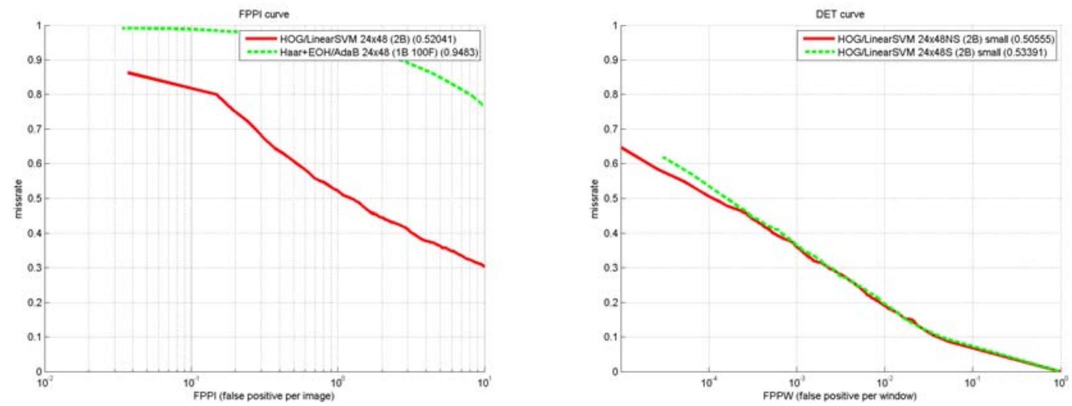


Figure 14: Left: Per-image evaluation of the HaarEOH and HOG methods trained with small samples and tested in images with small pedestrians. Right: Difference between the use of small pedestrians scaled from big images versus actual small pedestrians.

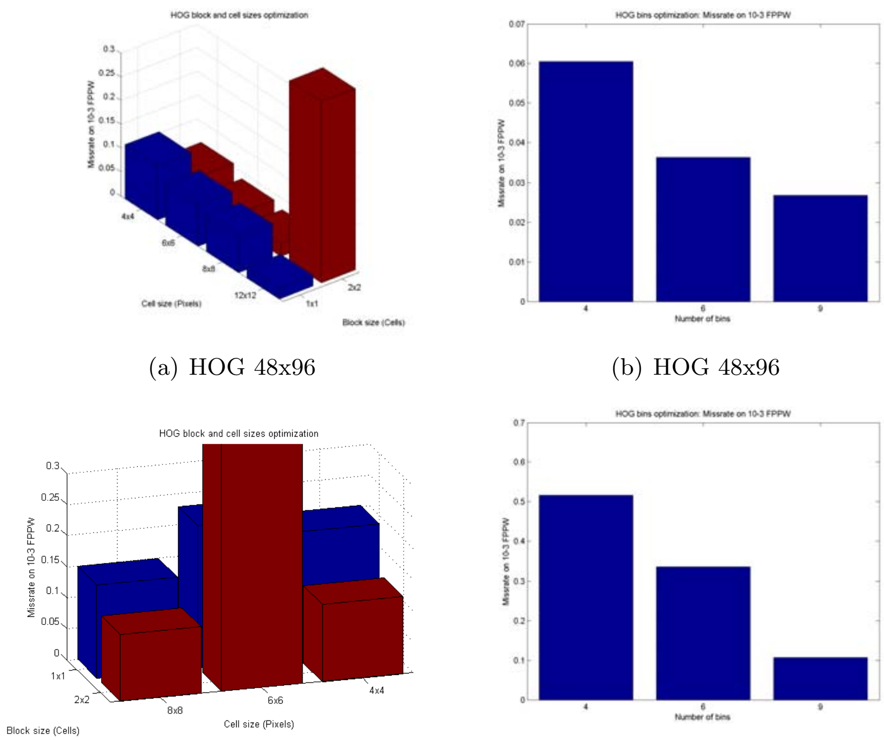
of block size (1x1 and 2x2 cells) and different cells sizes (4x4, 6x6, 8x8 and 12x12 pixels) over the small and big images of Daimler09 dataset. In Figure 15b and 15d we can also see the missrate at 10^{-3} FPPW for a HOG with the original configuration but different bins size (4, 6 and 9 orientations). Analyzing the results it seems that a fine binning and large scale features are better because they give more information than small features with less orientations. Also the best parameters of HOG are blocks of 2x2 cells of 8x8 pixels with 9 orientation bins for the three used databases: INRIA, Daimler09 small and Daimler09 big.

3.3 Discussion

At the beginning of the work raised some questions and after the experiments, some other questions appeared. Let's answer them.

• ***Which are the most suitable datasets for PPSs?:*** Although, the INRIA dataset is not the most suitable for training a pedestrian detector, we use it because it is the most spread and it allow us to compare with other authors. Recently, two new datasets the Caltech and the Daimler09 have been made public. After working and analyzing these datasets we have seen that the Caltech dataset has not enough different examples for training and testing, these examples are usually very small, partially occluded and not well centered. On the contrary, the Daimler dataset is more suitable for our purposes: it has many more examples, well labeled and at several scales. The problem of this dataset it is that the original frames from which the training samples were extracted are not available.

• ***How does affect the size of the training images in the performance?:*** The HOG descriptor works better for closer pedestrians and it does not work pretty well for far ones. This is because this descriptor looks for the details of the pedestrian and these details are not present in the small ones. However, surprisingly the HaarEOH works better with small images than with big ones. We think that this is due to: (1) the features chosen by the AdaBoost are global and use to cover big areas like the hole body, the legs or the head and this big parts are still visible for far pedestrians and (2) the EOH features, comparing



(a) HOG 48x96 (b) HOG 48x96

(c) HOG 24x48 (d) HOG 24x48

Figure 15: HOG parameter optimization in the Daimler09 dataset over 48x96 and 24x48 images.

the density of two orientations in a region, try to describe the contour of some areas of the pedestrians but when the pedestrians are big the texture of the clothes introduce noise that can occlude the important edges.

• ***In order to learn a classifier for far away pedestrians, do we need samples with small pedestrians for training or is it enough to downscale the big ones?:*** In the study of the effect of the distance, we lack of information to answer this question. We do not have training sets with enough examples at several scales, so the best we could do was to downscale the original examples. It would be important to have real training images at several scales because a far pedestrian is usually more blurred than a near pedestrian scaled to the same size. Of course, artificial blurring could be introduced but it could not be the same as the camera effect. Anyway, we have splitted the test set, that actually have small samples, in a new training and test sets. Then we have trained a classifier with actual small pedestrians and scaled ones and the results show that the classifier trained on the actual pedestrians is slightly better than the other. Thus, we expect that if we get a training set with more actual small pedestrians this performance difference could be higher.

• ***How does affect the size of the training images in the optimum parameters?:*** The HaarEOH parameters have not been optimized because they have been fixed to have the maximum number of features that allow a feasible training. After the study of the HOG parameters we can conclude that there is a set of canonical parameters that performs well for all the cases and they are not affected by the pedestrian size. However, as we have mentioned if we train an AdaBoost with the whole dataset and with more features the HaarEOH would perform better in the per-image evaluation.

- ***Which method performs better for each distance?:*** Analyzing the per-window results we can observe that HOG performs slightly better than HaarEOH for big images and HaarEOH performs better for small images. This is because HOG descriptor rely more on local information or details that tend to disappear in small images while HaarEOH descriptor rely more in global information that it is also present in small images. On the contrary, on the per-image evaluation the HOG method is the best for all the sizes.

- ***What are the differences between the performances obtained by the per-window and the per-image evaluation?:*** We have seen that the per-window performance could be not very realistic as it does not take into account the errors caused by the sliding window (scale and localization of the target or false positives over body parts) and the NMS. For instance, the HaarEOH method performs similar to the HOG in the per-window evaluation and it gives worse results in the per-image.

- ***Why the two methods performs similar in the per-window evaluation and so different in the per-image evaluation?:*** The HOG method have a better localization in the position and size of the pedestrian because it models the pedestrian contour in the SVM weights and try to fit them with the contour of the pedestrian to detect. However, the HaarEOH has a worst localization because it relies on more global features of the pedestrian. Although the HaarEOH gives bad results in the per-image evaluation, we can see that these errors are due to localization problems and not because it gives false positives where there are not pedestrians like clutter backgrounds. This problem of localization with the HaarEOH could be solved using more features in the training phase to allow the learning machine to take also some local features.

- ***Should a pedestrian detection system take into account the pedestrian distance?:*** A typical system based on HOG is trained with images of the smaller size that should be detected. As the HOG method performs better as bigger are the training images, the most intelligent strategy should be to create multiple detectors specialized in different ranges of sizes. To detect small pedestrians with a sliding window approach is very time consuming because there is needed to scan thousands of windows. Then, if we need to detect these pedestrians in real time HaarEOH should be used.

4. Conclusions

Pedestrian detection is a very challenging problem that is not still solved and there are some aspects that are interesting to explore like the datasets used for learning the classifiers or the effect of the distance in the detection. The appearance of pedestrians varies significantly as a function of distance and those pedestrians that are close to the camera are seen with more detail than the ones which are further away. This leads us to think that a system based on multiple classifiers specialized on different depths could improve the overall performance with respect to a typical system based on a single general detector.

We explore this effect in order to answer some questions like which datasets are the most suitable for training a pedestrian detector, how the size of the training samples affects the performance of the pedestrian detectors and their optimum parameters, which method performs better for each distance and, the most important, whether it is necessary to use different classifiers specialized to different distances or it is enough to have a single classifier for every distance.

In order to solve these questions we have explored and analyzed different datasets and pedestrian detection methods. The available datasets have been analyzed deeply to choose the most suitable for our experiments and the selected ones are INRIA and Daimler09. Among the pedestrian detectors the HOG+SVM and HaarEOH+AdaBoost have been selected and implemented to understand their behavior. The two pedestrian detectors have been trained and evaluated with two different approaches (per-window and per-image) on two datasets for two different sizes (small and big).

Several experiments have been done and their results have been analyzed in depth to solve the initial questions and some other that raised during the experiments. Then a discussion about these questions have been done and we have realized that the distance is of key importance in pedestrian detection system. Finally we propose to build a system that combines specialized classifiers optimized for different ranges of sizes to improve its performance and computational cost.

Therefore, the contributions of this work are three fold. First, we analyze the latest datasets that have not been explored in detail. Second, we explore the behavior of several pedestrian detection approaches in such datasets. Third, we study the effect of the distance in pedestrian detection, including the step of parameter tuning for the different distances. In addition, a general discussion of the effect of the distance is also given.

As future work it would be interesting to include temporal features and tracking to study the effect of combining classifiers specialized in different distances. We think that such strategy would allow to detect pedestrians from further distances and would lead to a more robust system.

References

[1] P. Marchal D. Gavrila and M.-M Meinecke. SAVE-U, deliverable 1-a: Vulnerable road user scenario analisys. Technical report, Society Technology Programme of the EU, 2003.

[2] David Gerónimo, Antonio M. López, Angel D. Sappa, and Thorsten Graf. Survey on pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[3] Markus Enzweiler and Dariu M. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(1), 2008.

[4] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, November 1995.

[5] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics).* Springer, November 1999.

[6] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag.

[7] D. M. Gavrila, J. Giebel, and S. Munder. Vision-based pedestrian detection: The protector system. In *In IEEE Intelligent Vehicles Symposium*, pages 13–18, 2004.

[8] Liang Zhao and Chuck Thorpe. Stereo- and neural network-based pedestrian detection. *IEEE Trans. on Intelligent Transportation Systems*, 1:148–154, 1999.

[9] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *Int. J. Comput. Vision*, 38(1):15–33, 2000.

[10] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. pages 511–518, 2001.

[11] David Geronimo, Angel D. Sappa, Antonio Lopez, and Daniel Ponsa. Adaptive image sampling and windows classification for on-board pedestrian detection. 2007.

[12] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893, 2005.

[13] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[14] Oncel Tuzel, Fatih Porikli, and Peter Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10):1713–1727, 2008.

[15] P. Felzenszwalb, D. Mcallester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Anchorage, Alaska, June 2008.*, June 2008.

[16] Ponce1, T. L. Berg3, M. Everingham4, D. A. Forsyth1, M. Hebert5, S. Lazebnik1, M. Marszalek6, C. Schmid6, B. C. Russell7, A. Torralba7, Williams8, Zhang6, and A. Zisserman4. Dataset issues in object recognition.

[17] Bo Wu and Ram Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 90–97, Washington, DC, USA, 2005. IEEE Computer Society.

[18] Bo Wu and Ramakant Nevatia. Cluster boosted tree classifier for multi-view, multipose object detection. In *ICCV*, pages 1–8, 2007.

[19] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009.

[20] S. Munder and D.M. Gavrila. An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1863–1868, 2006.

[21] kobi Levi, Yair Weiss, and Of Good Features. Learning object detection from a small number of examples: the importance of good features, 2004.

Acknowledgments

I would like to acknowledge my Advisors Dr. Antonio López and David Gerónimo. Also to Dr. Dani Ponsa for the implementations of Haar descriptor and RealAdaBoost and to David Gerónimo for his implementation of HOG and EOH.